

BEhaviour of ELephants in Dublin zoo

**Final Report**

Higher Diploma in Science in Data Analytics

Supervisor: Paul Laird

Student: Beth Craig (10331736)

10331736@mydbs.ie

22/08/2018

# Contents

Abstract [0](#_Toc522552352)

**[Contents](#_Toc522552353)** [1](#_Toc522552353)

[1. Introduction 2](#_Toc522552354)

[2. Requirements Specification and Design 3](#_Toc522552355)

[2.1 Requirements 3](#_Toc522552356)

[2.2 Design 3](#_Toc522552357)

[3. Implementation 3](#_Toc522552359)

[3.1 Software 3](#_Toc522552360)

[3.2 Hardware 5](#_Toc522552361)

[3.3 High Level Implementation Decisions 6](#_Toc522552362)

[3.4 Implementation of the Requirements 6](#_Toc522552362)

[4. Implementation 11](#_Toc522552395)

[5. Testing and Results 11](#_Toc522552395)

[6. Conclusion and Future Work 12](#_Toc522552398)

[7. References / Bibliography 13](#_Toc522552401)

**[Appendix A](#_Toc522552404)** [Supporting files 14](#_Toc522552404)

**Abstract**

The cleaning, transformation and statistical analysis of data from a study to determine the effects of changing the feeding approach for Elephants in Dublin Zoo was automated using Python. Dashboards were created in Tableau to visualize the output and provide the end user with a tool for exploratory data analysis. The study was found to be inconclusive due in part to issues in the implementation. The type of feeding was confounded with other possible contributing factors such as day of the week and whether there was training on a given day.

**1.** **Introduction**

Dina is the matriarch and mother and grandmother to most of the herd of elephants in Dublin Zoo. In 2017, animal behaviourists were consulted by the zoo due to concerns over a specific behaviour that Dina was displaying, referred to as ‘stereotyping’. Stereotypic behaviour is defined as behaviours which are repetitive, unvarying and have no apparent goal or purpose (Elzanowski, 2006). In Dina’s case she would sway from side to side repeatedly for long (and sometimes short) periods of time.

After an initial assessment, the animal behaviourist advised that this behaviour was most likely a result of a combination of factors in Dina’s early life prior to arriving in Dublin Zoo including early weaning, physical restriction and repeated breaking of social ties when Dina was moved from one zoo to another (Latham, 2008).

It was hypothesised that a change to the way Dina and the other elephants are fed to more closely mimic how elephants feed in the wild could reduce the amount of swaying behaviour. This change involved using fresh branches of trees delivered in a variety of locations in the elephant enclosure instead of hay from a bail. This style of feeding is closer to an elephants natural feeding pattern and provides recreation and activity as they have to work harder to obtain their required calorie intake (Rees, 2009).

This change to feeding was trialled over a few days to test the feasibility of the suggestion. A reduction in swaying was observed and so a larger study was planned. Using CCTV footage, Dina’s movements between 5am and 8pm over a period of 4 weeks were catalogued as a series of behaviours from a pre-defined list. The start and finish time of each episode was recorded, along with some supplementary details including location within the elephant enclosure and any activities underway. The elephants were fed the ‘normal’ way for the 1st and 4th week and the alternative ‘fresh’ way for the 2nd and 3rd week.

The primary purpose of this project was to analyse the output of this study to (i) characterize Dina’s behaviours as % time spend at a behaviour in 15 minute intervals (specifically requested by the zoo) (ii) determine if the change in feeding approach reduces the amount of swaying and (iii) communicate the results to different groups including the zoo keepers and zoo management and prepare the analysis for publication in a peer reviewed journal. A secondary purpose was to generalise the solution so that it could be used for other behavioural intervention studies.

**2. Requirements Specification and Design**

**2.1 Requirements**

The following list of high level requirements were formulated with the animal behaviourist

**Primary Objectives**

2.1.1 Clean the data

2.1.2 Automate the transformation of the data from start and finish times to the % time spent at each behaviour in 15 minutes intervals

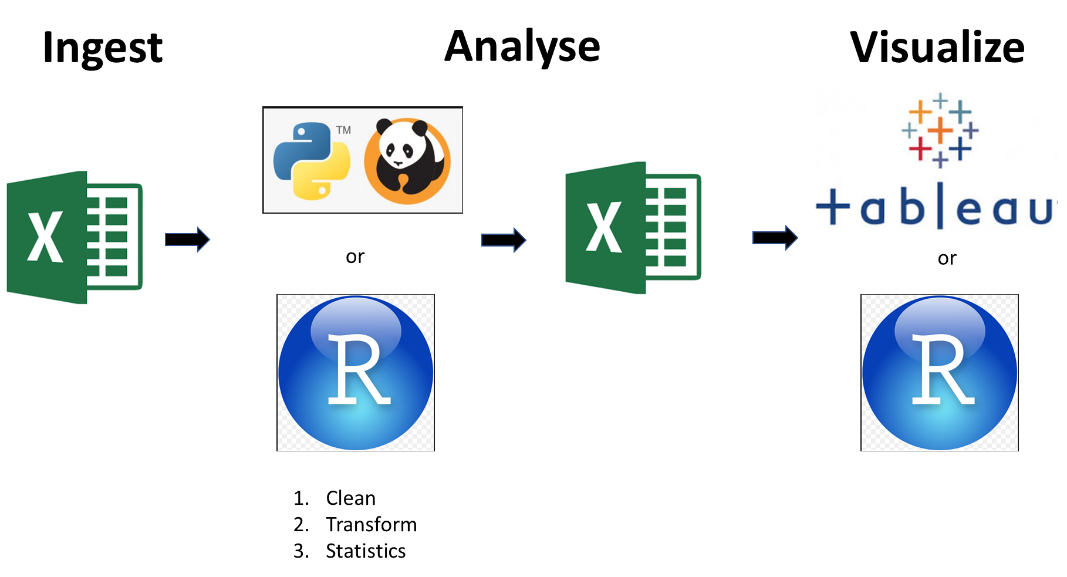
2.1.3 Determine if the change in feeding approach caused a statistically significant change in the % time spent swaying (alpha = 0.05)

2.1.4 Create dashboards to enable self-service exploratory analysis and to communicate the results to the management of the zoo and zoo keepers

**Secondary Objective**

2.1.5 Create a pipeline and generalize the solution so that it can be reused for other intervention studies

**2.2 Design**

**3.2.1 High Level Design**

## **Figure 1** High level design. Data needs to be ingested from excel into either R or Python where it can be cleaned, transformed and statistics performed. The output will then be written to csv where it can be ingested by Tableau or R for visualization.

Refer to **Figure 1** for an overview of the general design approach. The data is already in a csv file. It needs to be ingested into a programming environment (either Python or R) to be cleaned, transformed and the required statistics carried out. The transformed data needs to be visualized and interactive plots created to enable self-service exploratory analysis.

**3. Implementation**

## **3.1 Software**

Refer to **Table 1** for the software used and the version.

|  |  |
| --- | --- |
| Software | Version |
| Excel | 2016 |
| Python | 2.7 |
| Minitab | Minitab 18 |
| Anaconda Navigator | 1.5 |
| Jupyter Notebook | 4.3.1 |
| Tableau | 2018.2 |

**Table 1** Software

## **3.2 Hardware**

All work was completed using a Lenovo Yoga 710 with an Intel Core i7 (7th Gen) processor.

## **3.3 High Level Implementation Decisions**

The overall objective of this project was to produce a tool that meets the requirements set out above with as little input from the end user as is possible.

The major high-level decisions were (i) whether to use R or Python or a combination of both for the analysis section and (ii) whether to use R or Tableau for the visualizations.

There is a package in R (reticulate), that allows a user to call and execute python script within R and returns the output into the R environment as the corresponding R objects. This was trialled and worked but ultimately, I decided to up skill and learn how to code the statistics in Python as this simplified the process and provided a good learning opportunity for me.

I decided to use Tableau for the visualizations due to the user-friendly environment for self-service data exploration and the time required to teach myself how to do it using Python or R.

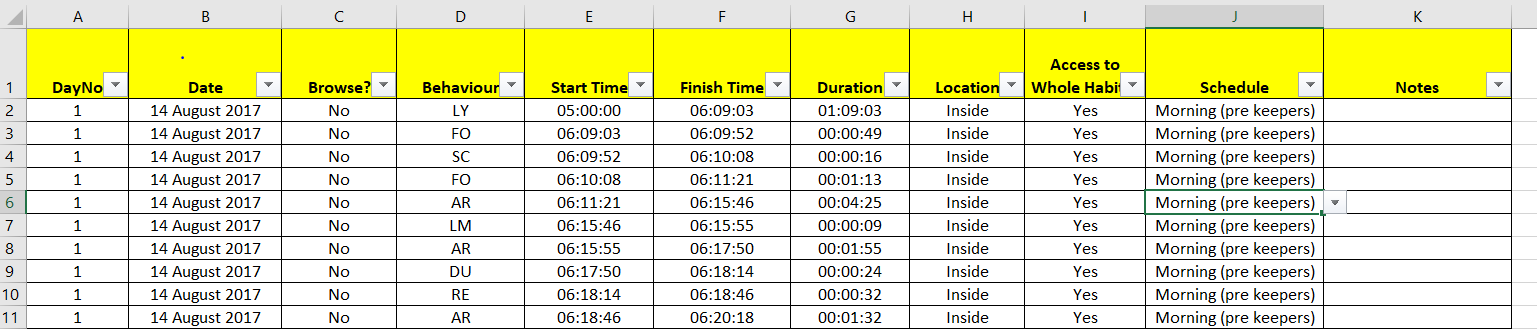
**3.4 Implementation of the Requirements**

**3.4.1 Clean the Data**

**3.4.1.1 Overview of the Data**

Data was originally captured using excel templates with a new file generated for each day of the study (sometimes more than one). The template exerted a certain amount of control over data entry (eg drop down lists for the behaviour) but a number of different versions of the template were used and data was entered by circa 20 students.

At the start of this study, the individual files had already been compiled into a master data csv and had been inspected manually for impossible and/or typographical entries. There are 11 variables with 7529 observations in total. Refer to **Snip 1**. Refer to **Table 2** for a description of the variables.



**Snip 1** Screen shot of the master csv file

|  |  |
| --- | --- |
| **Variable Name** | **Description** |
| DayNo | Study day number. Integers starting from 1 |
| Date | The calendar date ‘dd Month yyyy’ |
| Browse? | Binary (Yes for browse days, No for normal) |
| Behaviour | List of pre-defined behaviours |
| Start Time | Time a behaviour is recorded to start (hh:mm:ss) |
| Finish Time | Time a behaviour is recorded to finish (hh:mm:ss) |
| Duration | Difference between Finish and Start Time (hh:mm:ss) |
| Location | Binary (Inside/Outside) |
| Access to Whole Habitat | Binary (Yes/No) |
| Schedule | List of pre-defined activities that describes a typical day |
| Notes | Text, input at user discretion |

**Table 2** Name and Description of each of the original 11 variables

During the study there were issues with the CCTV cameras on two days (days 9 (22nd August) and 10 (23rd August)). These two days were removed from the study and it was extended by two days to make up for the loss of information.

In conversation with the zookeepers it was revealed that enough fresh feed was delivered for two days. So the type of feeding day was further identified as ‘Fresh’ (meaning the fresh food was delivered that day), ‘OneDayOld’ (fresh food available to the animal but delivered the day before) and ‘TwoDaysOld’ (fresh food available to the animal but delivered two days previously). There was only one instance of ‘TwoDaysOld’ and this was an error in the implementation of the study.

This information, along with other variables that were constant for a given day of the study (Date and Browse, DOTW) were captured in a meta data file and joined as required in Pandas and Tableau. Refer to **Snip 2**.

**A picture containing crossword puzzle

Description generated with high confidence**

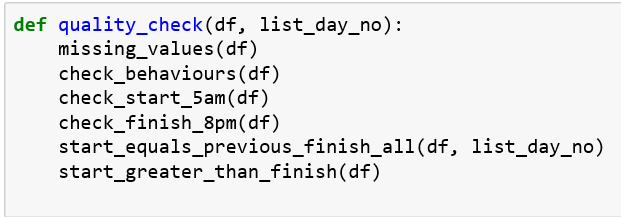
**Snip 2** The metadata file. Variables that were constant for a given day of the study were joined with the augmented ‘Day Type’ variable and joined with the resto f the data in Pandas and Tableau as required.

**3.4.1.2 Functions to check for errors**

Functions were written in Python to

1. Check for missing data
2. Check that every day started at 5am
3. Check that every day finished at 8pm
4. Check that within a day, every start time except for the first one equals the previous finish time
5. Check that for every pair of start and finish times, the start time was less than the finish time
6. Check that only behaviours from the pre-defined list were used.

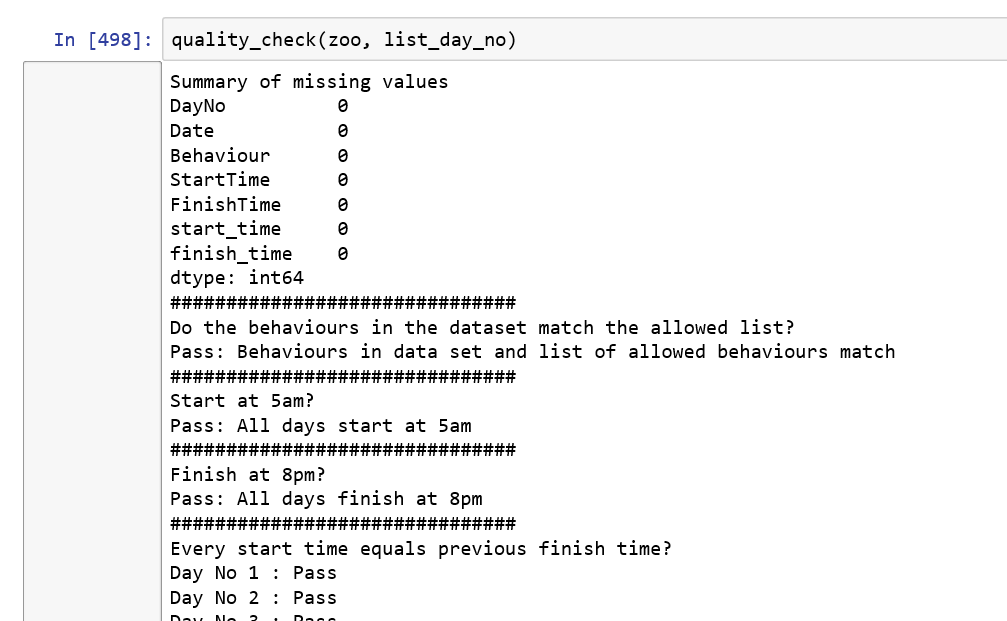
These individual functions were rolled into one called **quality\_check.** Refer to **Snip 3**.



**Snip 3** Individual quality checks were rolled into one function to simplify the process for the end user. The data frame (df) and list of day numbers to be checked (list\_day\_no) are passed to the function quality\_check().

The functions return a print out of any issues detected. Refer to **Snip 4**. It is up to the end user to implement a solution. This is in part necessary because the CCTV footage needs to be revisited to find out what was supposed to have been entered. But future iterations could include some options for automation of error correction. For example, if the problem is that a behaviour as recorded past 8pm, the end-user could simply choose to truncate the data at 8pm.

Despite the initial manual inspection of the data, the quality\_check() function revealed numerous issues with the raw data, which were rectified in consultation with the team in the zoo. These included behaviours being recorded past 8pm, new observations starting before the finish time of the previous one, missing seconds (start time > previous finish time) and typographical errors for Behaviour.



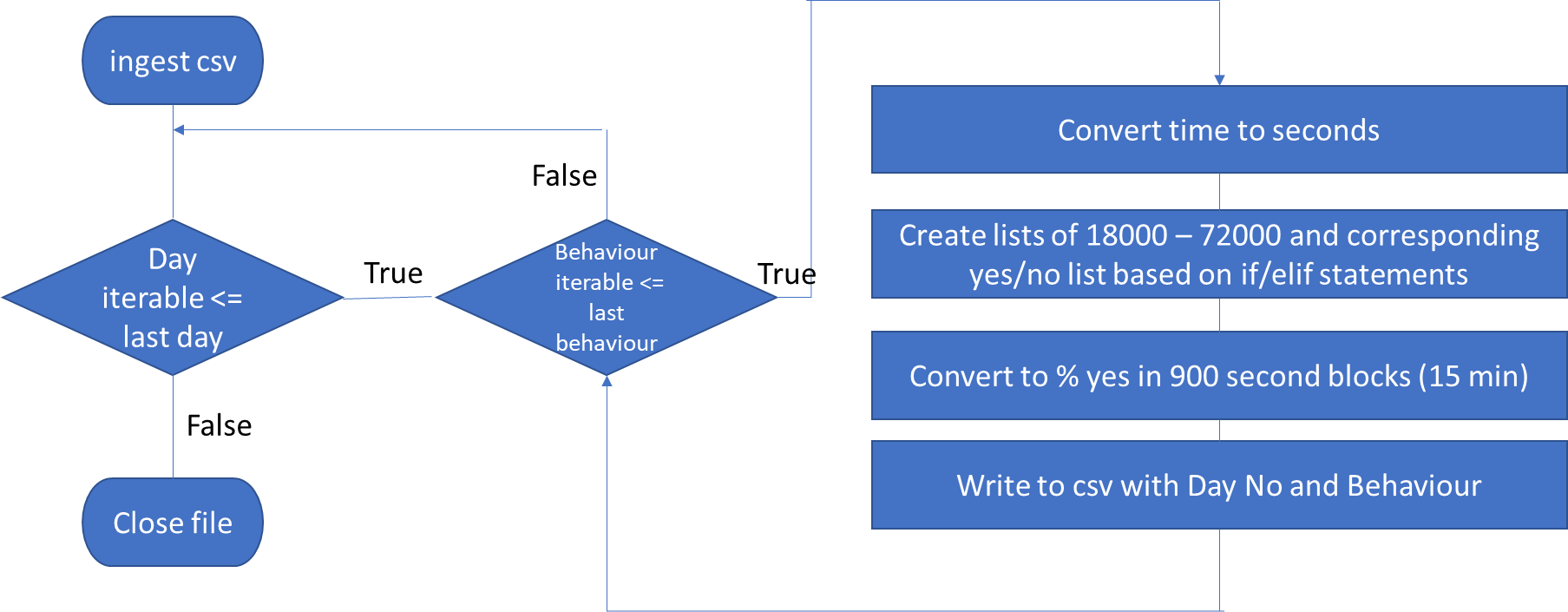
**Snip 4** Output from the quality\_check() function. In this example, there are no issues detected.

**3.4.2 Automate the transformation of the data from start and finish times to the % time spent at each behaviour in 15 minutes intervals**

## **3.4.2.1 Proof of Concept Algorithm in Python**

A proof of concept algorithm was developed in Python to convert the collected raw data (refer to **Snip 1**) to the required format (% time spent at a behaviour in 15-minute intervals).

Using a list for each variable, each of the 54,000 seconds between 5am (18,000th second) and 8pm (72,000th second) was designated either Yes or No for each behaviour using conditional statements. After filtering for a given behaviour on a given day, the algorithm looped through the start and finish times lists and using conditional statements, created a new corresponding list with a ‘Yes’ for every second where the particular behaviour is recorded to have occurred and a ‘No’ when it is not recorded to have occurred. The algorithm then repeated the process for every other behaviour in the list and then repeated the entire process for each day (ie behaviour was nested within day). The % of time spent at a behaviour (%TSB) was then calculated by counting how many ‘Yes’ s there were in every 900 seconds and then simple arithmetic to calculate the %. Refer to attached file ‘proof\_of\_concept\_090718’ for the code.



**Figure 2** Diagram of the proof of concept algorithm to convert start and finish times to the % time spent (%TSB) at a behaviour in 15 minute intervals

**3.4.2.2 Improving the Run Time**

**3.4.2.2.1 Using Line\_Profiler to identify hot spots**

While the proof of concept algorithm worked, it was slow. It took almost eight hours to complete the transformation. The Line\_Profiler module was installed and used to determine if there were particular functions or lines within functions causing the slow progress. A subset of the data (DayNo 1,2,3) was used for this analysis to reduce the runtime.

Of the 4 custom built functions, 1 function **(yes\_no\_every\_second()**) was accounting for 99% of the entire run time. Within this function 1 of the 23 lines (**if i not in time:** ) was accounting for 86% of the time. This time was not explained by the number of hits but instead a 1000 fold increase in time per hit compared to the other lines in the function.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Function name | Run time | % total run time |
| 1 | convert\_time | 0.00110334 | 0.01 |
| 2 | create\_lists | 0.0517941 | 0.06 |
| 3 | yes\_no\_every\_second | 86.1443 | 99.03 |
| 4 | convert\_TSB\_percent | 0.790801 | 0.91 |

**Table 3** Summary of the result of using line\_profiler to identify how long each of the 4 functions used in the proof of concept algorithm were taking.

This line was replaced with if/elif statements to implement the same logic. Although this change increased the number of lines in the code, it reduced the run time of the function by 70%, resulting in a corresponding ~ 70% reduction in run time for the entire programme.

**3.4.2.2.2 Pandas in a Jupyter notebook**

To further reduce the time to complete and improve the usability of the code, the algorithm was translated into Pandas and a Jupyter notebook.

For the most part, the same logic was employed but use was made of Pandas in built functions and capabilities such as resampling and filtering by column values to avoid loops and lists as much as possible. Refer to XX for the code.

Using the same data subset (DayNos 1,2 and 3), the run time of the Pandas code was only minutes, with the entire 28 days (7529 observations) taking only 12 minutes to complete, a reduction of 98% on the first proof of concept algorithm and well within an acceptable run time for the zoo.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Attempt | Run Time (hours) | Run Time  (min) | % Reduction |
| 1 | 1st POC algorithm | 8 | ~480 | 0 |
| 2 | POC after line\_profiler | 2.2 | ~140 | 70 |
| 3 | Pandas | 0.2 | 12 | 98.5 |

**Table 4** Summary of the 3 main iterations of the code to transform the data to the required % TSB format.

**3.4.2.2 Function to check correct implementation of % TSB**

A function called check\_6000() was implemented as part of the final iteration of this code to ensure that the sum of %TSB for each interval is 100% and for each day is 6000% (4 intervals per hours for 15 hours = 60).

**3.4.3 Determine if the change in feeding approach caused a statistically significant change in the % time spent swaying (alpha = 0.05)**

The required statistics were included in the Pandas Jupyter notebook script along with the cleaning and transformation to %TSB.

The %TSS (% time spent swaying) per day of the study was calculated as part of the python script and joined the metadata csv. The %TSS for browse and non-browse was assessed for normality using normality plots and the Shapiro-Wilk test. Bartlett’s test was used to confirm the assumption of equal variance.

The difference in the mean %TSS for browse and non-browse days was formally tested using the **stats.ttest\_ind()** command. A one-way Anova with 4 factors was implemented using **stats.f\_oneway ()**. Tukey comparisons were generated by calling the **tukeyhsd()** function after **multi.MultiComparison()**.

The process was repeated with one suspected outlier day removed and also with the data from the feasibility study included.

**3.4.4 Create dashboards to enable self-service exploratory analysis and to communicate the results to the management of the zoo and zoo keepers**

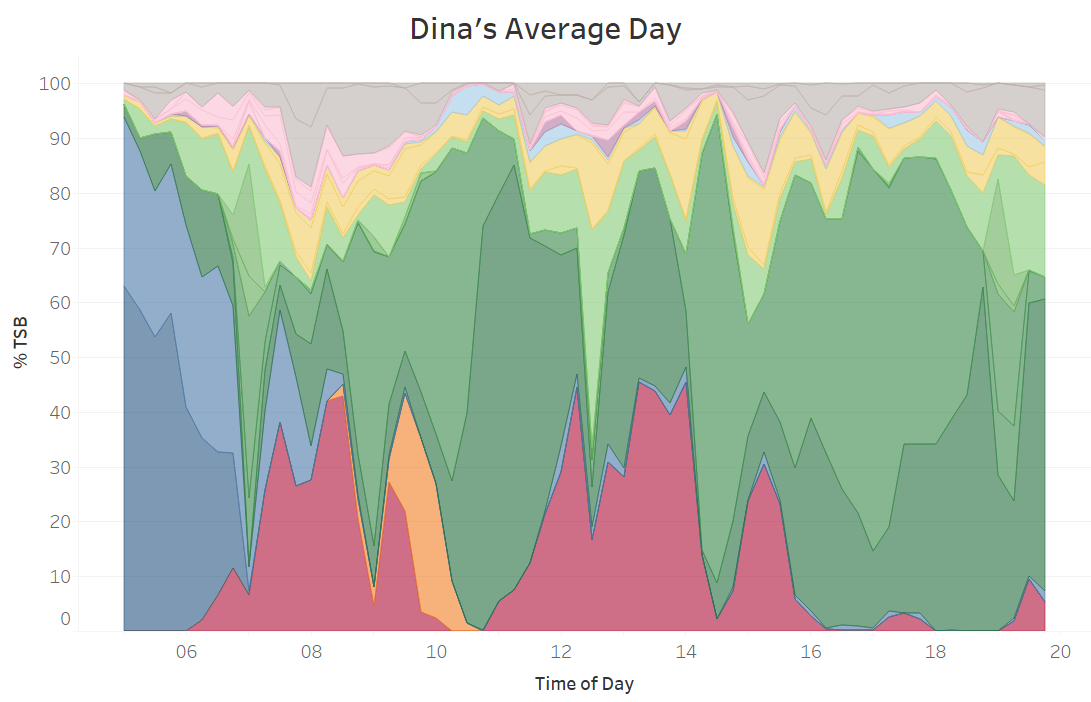
A major task of this project is to visualise Dina’s day in a way that her behaviour can be easily understood by a variety of people (vets, zookeepers, management). Refer to **Graph 1** for the selected type of visualization, produced in Tableau. **Graph 1** is Dina’s average day, meaning the average for %TSB over all the days in the study.

The % time spent at a given behaviour is on the y-axis with time of day on the x-axis. The different categories of behaviour are colour coded (sleeping/resting is in dark blue for night time, swaying is in red as it is undesirable, feeding behaviours are different shades of green, drinking water is light blue etc). The width of the band is proportional to the % of time in each 15-minute block spent at a given behaviour. Browsing is the darkest green and is stacked directly above swaying (red) to allow for easy visual comparison of the two. The different shades of grey represent times when it was not possible to decipher Dina’s behaviour (off camera, camera not working etc) and these are stacked at the top, so they are separated from behaviours of interest.

These graphs show that there are two distinct clusters of swaying, one in the morning from around 7am to 9.30am and one in the afternoon from around 12pm to 4pm and that training (off white) is what separates the two clusters. After 4pm in the evening, very little swaying occurs. The reduction in swaying on the fresh feeding days is a result of a reduction in the % of time Dina sways (mostly) in the afternoon as opposed to eliminating periods of swaying.

Another observation is that Dina always stops swaying just after 2pm for a short period regardless of the feeding approach or day of the week.

In consultation with the end users, a dashboard with two identical replicas of this area graph with all relevant filter options was built. Each of the two graphs can be filtered separately to allow for direct comparisons between scenarios. For example, normal vs fresh, or weekend vs weekday. They can be set to a single date or multiple subsets of dates, effectively giving the end user complete control over slicing and dicing options.



**Graph 1** Dina’s Average Day is the average of all %TSB values for all days in the study. The % time spent at a given behaviour is on the y-axis with time of day on the x-axis. The different categories of behaviour are colour coded: Swaying (red), Feeding (green), Training (orange), Resting/Sleeping (blue), Locomotion (yellow), Drinking (light blue), other (pink/purple), unknown (grey).

**5. Testing and Results**

**5.1 Testing**

Testing throughout the building phase of the project was carried out using subsets of the dataset with known properties calculated in excel. In the Pandas script, functions to check for data quality were in built into the script (refer to **Section 3.4.1.2**). The script also includes a check that the sum of every 15 minute interval on every day sums to 100% and that the total for each day sums to 6000% (4 intervals for each of 15 hours = 60 intervals at 100% each). Refer to **Section 3.4.2.2**

Statistics were cross checked with the output from Minitab.

The area graphs in Tableau also provided a visual check that time intervals sum to 100%.

**5.2 Results**

**5.2.1 Exploratory Analysis**

Exploratory analysis was carried out in Tableau. Box-plots were created for the %TSS versus the feeding type (browse vs no browse and also normal, fresh, one day old and two days old) and DOTW. The %TSS was also plotted as a time series. Refer to **Graphs X** to **X**.

The most striking observation is that there is one out of trend Fresh day with a %TSS of 25% compared to the next highest of 11.2%. This is the only Fresh day implemented at the weekend.

Also evident is a trend in the DOTW boxplot. On average Saturday, Sunday and Monday have higher % swaying than Tuesday to Friday inclusive.

The area graphs and dashboard (refer to **Section 3.4.4**) were also used. These visualizations show that Dina’s swaying generally occurs in two cluster, one in the morning from about 7am to 9.30am and one in the afternoon from 11am to 4pm, with the browse feeding having the biggest effect on the afternoon swaying cluster.

It also shows that Dina rarely sways between 10 and 11 am. When she has training, this is when it is usually scheduled, but she doesn’t not sway during this time, even when no training occurs.

She also never sways at 2.30pm.This corresponds to the lowering of a hay net.

She rarely sways after 4pm which corresponds to gaining access to the Elephant house and a fresh supply of food.

**5.2.X Statistical analysis to determine if the change in feeding approach caused a statistically significant change in the % time spent swaying (alpha = 0.05)**

**5.2.1 t-test for difference between mean %TSS for browse and non-Browse days**

This task was complicated by the way the study was implemented.

An independent t-test was run to see if, despite the differences in the implementation of the browse days (see **Section 3.4.1.1**), there was a statistically significant difference between the average amount of swaying on browse days versus normal (no-browse) days.

Both subsets were evaluated using normality plots and the Shapiro Wilk test to confirm the validity of the assumption that the data is normally distributed.

For both subsets, the p-value of the Shapiro-Wilk test was > 0.05 and the normality plots were approximately a straight line. Bartletts test was used to test if they could be considered to have equal variance. It had a p-value > 0.05 (actual value 0.20).

The actual difference in the mean %TSS for browse and non-browse days was only 2.53%, giving a p-value of 0.25. We fail to reject the null hypothesis of a difference in the means of %TSS for browse and non-browse days.

**5.2.2 One-way Anova with 4 factors**

To check to see if the way the browse days were implemented had a statistically significant effect of the %TSS, a one-way Anova with 4 factors (normal, fresh, one day old and two days old) was also implemented. This was also not significant, with a p-value of 0.49.

In spite of the statistically insignificant result of the Anova, Tukey comparisons were also carried out to gain a deeper understanding of the differences, if any, between the different feeding types. Refer to **Snip X** for the code and the output.

A screenshot of text

Description generated with very high confidence

**Snip X** Code and output for the Tukey comparisons between the different feeding types.

Based on exploratory data analysis, the Tukey comparisons were repeated with one Fresh day (Sunday 3rd September, DayNo 21) removed. With this day removed, the difference between the Normal and Fresh groups is significant. Refer to **Snip X**. However, no reason for the out of trend result was determined.

A screenshot of a cell phone

Description generated with very high confidence

**Snip X** Output for the Tukey comparisons between the different feeding types with Sunday 3rd September (Day No 21) removed

**5.3 Critical Evaluation of the Final Project**

**5.3.1 Positive aspects**

* Speed of algorithm
* Automated report on the data quality
* Data cleaning, transformation and statistics in a single work flow
* Visualizations that make the entirety of the data collected easily accessible by stakeholders of all levels of technical understanding and none
* Jupyter notebook employed to facilitate ease of use end-users with no coding experience

**5.3.2 Shortcomings**

* No option to impute missing values/ correct incorrect entries within the script
* Pipeline – use of Tableau means system is not end to end
* Not generalised for use in future studies
* Statistics – Tableau required for exploratory analysis and to assess for possible outliers
* Statistics – hard coded for this study and for %TSS
* Not all of the available information is used (location, notes)

**6. Conclusion and Future Work**

**6.1 Review what the project achieved in terms of the proposed goals and project plan**

|  |  |  |
| --- | --- | --- |
| **#** | **Original Objective** | **Assessment** |
| **1** | Develop a proof of concept algorithm (get it to work) in Python to convert the data to the required % time spent at each behaviour in 15 minute intervals | Complete |
| **2** | Iteratively upgrade the proof of concept python script to reduce run time and improve usability | Complete   * Runtime reduced to 12 minutes from 8 hours * Use of Jupyter notebook to improve usability |
| **4** | Analyse the data using R to determine if any change in swaying due to change in feeding style is statistically significant (alpha = 0.05) | Complete   * Python used instead of R to simplify the process |
| **5** | Analyse the data for any other useful insights | Complete |
| **6** | Visualize as required | Complete   * Self-service dashboard in Tableau |
| **7** | Prepare results for presentation to zoo keepers, zoo management and publication | Partially Complete   * Dashboard prepared * Presentation made to zoo * Publication uncertain due to inconclusive results |
| **8** | Develop a pipeline of a generalised version of the python and R scripts to allow them to be re-used in future studies | Partially Complete   * Cleaning, transformation and statistics completed in a single script * Visualizations separate in Tableau * Not generalised for use in future studies due to time limitations |

**6.2 Changes from the interim report**

The main changes from the interim report are the decision to use Python instead of R for statistical analysis and to use Tableau instead of R for visualization.

The decision to use Python was to simplify the process as much as possible and in part a decision to focus on one programming language. The decision to use Tableau for visualization was in part due to time constraints and in part due to the skill set of the end users, who are familiar with Tableau and so can use it for self-service exploratory analysis.

Generalization of the script to allow it to be re-used for future studies was a time dependent secondary objective of this project.

**6.3 Reflect on the learning experiences gained in doing the project and its relevance to on-going progress as a learner and future practising IT professional**

Refer to **Table X** for the original stated learning objectives from the project proposal. All the objectives were chosen with future employment prospects in mind. Within the skills required of a data scientist, coding is the one which I had little experience of starting this course and so I chose a project that would enable me to spend time improving my ability to programme.

|  |  |  |
| --- | --- | --- |
| **#** | **Original Learning Objective** | **Assessment** |
| **1** | Build an algorithm from scratch | Complete |
| **2** | Learn how to use Pandas and Numpy | Complete |
| **3** | Learn how to use Jupyter notebooks | Complete |
| **4** | Improve understanding of and ability to use Github | Partially Complete |
| **5** | Develop R skills and the statistics required | Decision to use Python instead due to feedback on relevance of R and Python to Data Science employment opportunities in Dublin |
| **6** | Learn how to develop a pipeline so data can be piped through python, R and other tools to generate required output with minimized user input | Not achieved |
| **7** | Story telling/communication of results to different audiences | Feedback from end users on graphs is very positive  Easilt digested and understood by all audiences |

**6.4 Future Work**

The data was cleaned and exploratory analysis carried out. An algorithm has been put together and a proof of concept script generated in Python, converting the data to the format required for analysis and visualization. Preliminary analysis and visualization have been carried out in Minitab and Tableau. The next steps are to automate the statistical analysis and visualization using R or Python and then to iteratively improve the scripts to (i) reduce the run time, (ii) generalise and improve robustness and usability of the code and (iii) pipe the different phases to reduce the need for user input. The main challenges to date relate to my own stated learning objectives, namely to improve coding skills and learning how to use tools for communicating technical projects (Jupyter notebooks and Github). These will be more important in the second half of the project and will be prioritised accordingly. The run time for the proof of concept algorithm needs to be reduced and a robust testing regime established to ensure the accuracy of the output.

1. **References / Bibliography**

Craig, B. (2018) Behaviour of Elephants in Dublin Zoo, Proposal. [online] Available at <https://elearning.dbs.ie/mod/assign/view.php?id=539526> [Accessed 09 Jul. 2018].

Elzanowski, A. and Sergiel, A. (2006). Stereotypic behaviour of a female Asiatic elephant (Elephas maximus) in a zoo. *Journal of Applied Animal Welfare Science*, 9(3), pp. 223-232.

GitHub. (2018). [online] Available at: https://github.com/ [Accessed 12 Jun. 2018].

Jupyter.org. (2018). *Project Jupyter.* [online] Available at: http://jupyter.org/ [Accessed 12 Jun. 2018].

Latham, N. R. and Mason, G. J. (2008). Maternal deprivation and the development of stereotypic behaviour. *Applied Animal Behaviour Science*, 110(1), pp. 84-108.

Mango-solutions.com. (2018). *Integrating Python and R into a Data Analysis Pipeline - Part 1* [online] Available at: https://www.mango-solutions.com/blog/integrating-python-and-r-into-a-data-analysis-pipeline-part-1 [Accessed 12 Jun. 2018].

Pandas.pydata.org. (2018) Python Data Analysis Library — pandas: Python Data Analysis Library*.* [online] Available at: https://pandas.pydata.org/ [Accessed 12 Jun. 2018].

Python.org. (2018). Welcome to Python.org*.* [online] Available at: https://www.python.org/ [Accessed 12 Jun. 2018].

Rees, P. A. (2009). Activity budgets and the relationship between feeding and stereotypic behaviours in Asian elephants (Elephas maximus) in a zoo. *Zoo Biology,* 28(2), pp. 7997.

R-project.org. (2018). R: The R Project for Statistical Computing*.* [online] Available at: https://www.r-project.org/ [Accessed 12 Jun. 2018].

Tableau Software. (2018). [online] Available at: https://www.tableau.com/ [Accessed 12 Jun. 2018].